# DATA PREPARATION TECHNIQUES FOR WEB USAGE MINING IN WORLD WIDE WEB

**Sathiyamoorthi .V[1], Murali Bhaskaran .V[2]**
[1]Research Scholar/Lecturer/CSE Sri Shakthi Institute of Engineering and Technology, Coimbatore-62, Tamil Nadu, INDIA
[2]Principal, Paavai College of Engineering, Paachal-18, Tamil Nadu, INDIA
E-mail : [1]sathyait2003@gmail.com

## Abstract

The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites. The complexity of tasks such as Web site design, Web server design, and of simply navigating through a Web site has increased along with this growth. An important input to these design tasks is the analysis of how a Web site is being used. Usage analysis includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as finding the common traversal paths through a Web site.

Web site that has all the challenging aspects of real-life Web usage mining, including evolving user profiles and external data describing an ontology of the Web content. Even though the Web site under study is part of a nonprofit organization that does not "sell" any products, it was crucial to understand "who" the users were, "what" they looked at, and "how their interests changed with time," all of which are important questions in Customer Relationship Management (CRM).

Web Usage Mining is the application of data mining techniques to usage logs of large Web data repositories in order to produce results that can be used in the design tasks mentioned above. However, there are several preprocessing tasks that must be performed prior to applying data mining algorithms to the data collected from server logs. In this paper presents several data preparation techniques in order to identify unique users and user sessions. Also, a method to divide user sessions into semantically meaningful transactions is defined and successfully tested against two other methods. Here we are proposing an algorithm for knowing hidden information about web page access based on the previous access history stored in web log file.

**Keywords:** WWW, Preprocessing, Web Usage Mining

## I. INTRODUCTION

Web personalization has quickly moved from an added value feature to a necessity, particularly for large information services and sites that generate revenue by selling products. Web personalization can be viewed as using user preferences profiles to dynamically serve customized content to particular users. User preferences may be obtained explicitly, or by passive observation of users over time as they interact with the system. Principal elements of Web personalization include modeling of Web objects (pages, etc.) and subjects (users), matching between and across objects and/or subjects, and determination of the set of actions to be recommended for personalization. Existing approaches used by many Web-based companies, as well as approaches based on collaborative filtering, rely heavily on human input for determining the personalization actions. This type of input is often a subjective description of the users by the users themselves, and thus prone to biases. Furthermore, the profile is static, and its performance degrades over time as the profile ages. Recently, a number of approaches have been developed dealing with specific aspects of Web usage mining for the purpose of automatically discovering user profiles.

The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites. The complexity of tasks such as Web site design, Web server design, and of simply navigating through a Web site has increased along with this growth. An important input to these design tasks is analysis of how a Web site is being used. Usage analysis includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as finding the common traversal paths through a Web site. Usage information can be used to restructure a Web site in order to better serve the needs of users of a site. Long convoluted traversal paths or low usage of a page with important site information could suggest that the site links and information are not laid out in an intuitive manner.

The design of a physical data layout or caching scheme for a distributed or parallel Web server can be enhanced by knowledge of how users typically navigate through the site. Usage information can also be used to directly aide site navigation by providing a list of \popular" destinations from a particular Web page. Web Usage Mining is the application of data mining techniques to large Web data repositories in order to produce results that can be used in the design tasks mentioned above. Some of the data mining algorithms that are commonly used in Web Usage Mining are association rule generation, sequential pattern generation, and clustering. Association Rule mining techniques discover unordered correlations between items found in a database of transactions. In the context of Web Usage Mining a transaction is a group of Web page accesses, with an item being a single page access.

- 9.81% of the site visitors accessed the Atlanta home page followed by the Sneak peek main page.
- 0.42% of the site visitors accessed the Sports main page followed by the Schedules main page.

The percentages in the second set of examples are referred to as support. Support is the percent of the transactions that contain a given pattern. Both confidence and support are commonly used as thresholds in order to limit the number of rules discovered and reported. For instance, with a 1% support threshold, the second sequential pattern example would not be reported. Clustering analysis allows one to group together users or data items that have similar characteristics. Clustering of user information or data from Web server logs can facilitate the development and execution of future marketing strategies, both online and offline, such as automated return mail to visitors falling within a certain cluster, or dynamically changing a particular site for a visitor on a return visit, based on past classification of that visitor.

As the examples above show, mining for knowledge from Web log data has the potential of revealing information of great value. While this certainly is an application of existing data mining algorithms, e.g. discovery of association rules or sequential patterns, the overall task is not one of simply adapting existing algorithms to new data. Ideally, the input for the Web Usage Mining process is a file, referred to as a user session file in this paper that gives an exact accounting of who accessed the Web site, what pages were requested and in what order, and how long each page was viewed. A user session is considered to be all of the page accesses that occur during a single visit to a Web site. The information contained in a raw Web server log does not reliably represent a user session file for a number of reasons that will be discussed in this paper. Specifically, there are a number of difficulties involved in cleaning the raw server logs to eliminate outliers and irrelevant items, reliably identifying unique users and user sessions within a server log, and identifying semantically meaningful transactions within a user session.

This paper presents several data preparation techniques and algorithms that can be used in order to convert raw Web server logs into user session files in order to perform Web Usage Mining.

## II. RELATED WORKS

There are several commercially available Web server log analysis tools that provide limited mechanisms for reporting user activity, i.e. it is possible to determine the number of accesses to individuals and the times of visits. However, these tools are not designed for very high traffic Web servers, and usually provide little analysis of data relationships among accessed files, which is essential to fully utilizing the data gathered in the server logs. A maximal forward reference is the last page requested by a user before backtracking occurs, where the user requests a page previously viewed during that particular user session. For example, if a user session consists of requests for pages A-B-A-C-D-C, in that order, the maximal forward references for the session would be B and D. The links that are presented to a given user are dynamically selected based on what pages other users assigned to the same cluster have visited.

Two of the biggest impediments to collecting reliable usage data are local caching and proxy servers. In order to improve performance and minimize network traffic, most Web browsers cache the pages that have been requested. As a result, when a user hits the \back" button, the cached page is displayed and the Web server is not aware of the repeat page access. Proxy servers provide an intermediate level of caching and create even more problems with identifying site usage.

In a Web server log, all requests from a proxy server have the same identifier, even though the re- quests potentially represent more than one user. Also, due to proxy server level caching, a single request from the server could actually be viewed by multiple users throughout an extended period of time. Cookies are markers that are used to tag and track site visitors automatically. Another approach to getting around the problems created by caching and proxy servers is to use a remote agent. Instead of sending a cookie, sends a Java agent that is run on the client side browser in order to send back accurate usage information to the Web server. The major disadvantage of the methods is that rely on implicit user cooperation stem from privacy issues. There is a constant struggle between the Web user's desire for privacy and the Web provider's desire for information about who is using their site. Many users choose to disable the browser features that enable these methods.

## III. PREPROCESSING

Figure 1 how the preprocessing tasks of Web Usage Mining in greater detail than Figure. 2. The inputs to the

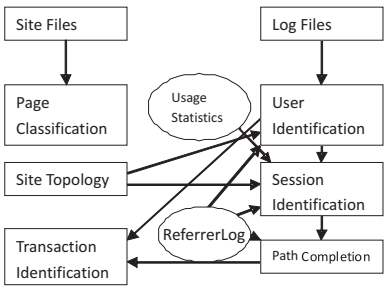



Fig. 1. Web Usage Mining Process

Preprocessing phase are the server logs, site files, and optionally usage statistics from a previous analysis. The outputs are the user session file, transaction file, site topology, and page classifications. As mentioned in Sect. 2, one of the major impediments to creating a reliable user session file is browser and proxy server caching. Current methods to collect information about cached references include the use of cookies and cache busting. Cache busting is the practice of preventing browsers from using stored local versions of a page, forcing a new down-load of a page from the server every time it is viewed. Cookies can be deleted by the user and cache busting defeats the speed advantage that caching was created to provide, and is likely to be disabled by the user. Another method to identify users is user registration, as discussed in section 2. Registration has the advantage of being able to collect additional demographic information beyond what is automatically collected in the server log, as well as simplifying the identification of user sessions. However, again due to privacy concerns, many users choose not to browse sites that require registration and logins, or provide false information.

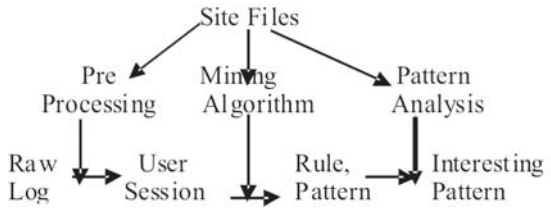



Fig. 2. Generalized Web Mining

*A. Data Cleaning*

Techniques to clean a server log to eliminate irrelevant items are of importance for any type of Web log analysis, not just data mining. The discovered associations or reported statistics are only useful if the data represented in the server log gives an accurate picture of the user accesses to the Web site. The HTTP protocol requires a separate connection for every file that is requested from the Web server. Therefore, a user's request to view a particular page often results in several log entries since graphics and scripts are down-loaded in addition to the HTML file. In most cases, only the log entry of the HTML file request is relevant and should be kept for the user session file. This is because, in general, a user does not explicitly request all of the graphics that are on a Web page, they are automatically down-loaded due to the HTML tags. Since the main intent of Web Usage Mining is to get a picture of the user's behavior, it does not make sense to include file requests that the user did not explicitly request. Elimination of the items deemed irrelevant can be reasonably accomplished by checking the suffix of the URL name. For instance, all log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG, and map can be removed. In addition, common scripts such as \count.cgi" can also be removed.

*B. User Identification*

Consider the following sample log log file

1. 123.456.78.9 - [25/Apr/1998:03:04:41 -0500] "GET A.html HTTP/1.0" 200 3290 - Mozilla/3.04 (Win95, I)
2. 123.456.78.9 - [25/Apr/1998:03:05:34 -0500] "GET B.html HTTP/1.0" 200 2050 A.html Mozilla/3.04 (Win95, I)
3. 123.456.78.9 - [25/Apr/1998:03:05:39 -0500] "GET L.html HTTP/1.0" 200 4130 - Mozilla/3.04 (Win95, I)
4. 123.456.78.9 - [25/Apr/1998:03:06:02 -0500] "GET F.html HTTP/1.0" 200 5096 B.html Mozilla/3.04 (Win95, I)
5. 123.456.78.9 - [25/Apr/1998:03:06:58 -0500] "GET A.html HTTP/1.0" 200 3290 - Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6. 123.456.78.9 - [25/Apr/1998:03:07:42 -0500] "GET B.html HTTP/1.0" 200 2050 A.html Mozilla/3.01 (X11, I, IRIX6.2, IP22)
7. 123.456.78.9 - [25/Apr/1998:03:07:55 -0500] "GET R.html HTTP/1.0" 200 8140 L.html Mozilla/3.04 (Win95, I)
8. 123.456.78.9 - [25/Apr/1998:03:09:50 -0500] "GET C.html HTTP/1.0" 200 1820 A.html Mozilla/3.01 (X11, I, IRIX6.2, IP22)
9. 123.456.78.9 - [25/Apr/1998:03:10:02 -0500] "GET O.html HTTP/1.0" 200 2270 F.html Mozilla/3.04 (Win95, I)
10. 123.456.78.9 - [25/Apr/1998:03:10:45 -0500] "GET J.html HTTP/1.0" 200 9430 C.html Mozilla/3.01 (X11, I, IRIX6.2, IP22)
11. 123.456.78.9 - [25/Apr/1998:03:12:23 -0500] "GET G.html HTTP/1.0" 200 7220 B.html Mozilla/3.04 (Win95, I)
12. 123.456.78.9 - [25/Apr/1998:05:05:22 -0500] "GET A.html HTTP/1.0" 200 3290 - Mozilla/3.04 (Win95, I)
13. 123.456.78.9 - [25/Apr/1998:0 5:06:03 -0500] "GET D.html HTTP/1.0" 200 1680 A.html Mozilla/3.04 (Win95, I)

Next, unique users must be identified. As mentioned previously, this task is greatly complicated by the existence of local caches, corporate firewalls, and proxy servers. The Web Usage Mining methods that rely on user

cooperation are the easiest ways to deal with this problem. However, even for the log/site based methods, there are heuristics that can be used to help identify unique users. Even if the IP address is the same, if the agent log shows a change in browser software or operating system, a reasonable assumption to make is that each different agent type for an IP address represents a different user. All of the log entries have the same IP address and the user ID is not recorded. However, the fifth, sixth, eighth, and tenth entries were accessed using a different agent than the others, suggesting that the log represents at least two user sessions. The next heuristic for user identification is to use the access log in conjunction with the referrer log and site topology to construct browsing paths for each user. If a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, again, the heuristic assumes that there is another user with the same IP address. Looking at the sample log again, the third entry, page L, is not directly reachable from pages A or B. Also, the seventh entry, page R is reachable from page L, but not from any of the other previous log entries. This would suggest that there is a third user with the same IP address. Therefore, after the user identification step with the sample log, three unique users are identified with browsing paths of A-B-F-O-G-A-D, A-B-C-J, and L-R, respectively. It is important to note that these are only heuristics for identifying users. Two users with the same IP address that use the same browser on the same type of machine can easily be confused as a single user if they are looking at the same set of pages. Conversely, a single user with two different browsers running, or who types in URLs directly without using a sites link structure can be mistaken for multiple users.

*C. Session Identification*

For logs that span long periods of time, it is very likely that users will visit the Web site more than once. The goal of session identification is to divide the page accesses of each user into individual sessions. The simplest method of achieving this is through a timeout, where if the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session. Many commercial products use 30 minutes as a default timeout, and established a timeout of 25.5 minutes based on empirical data. Once a site log has been analyzed and usage statistics obtained, a timeout that is appropriate for the specific Web site can be fed back into the session identification algorithm. This is the reason usage statistics are shown as an input to session identification in Fig. 4. Using a 30 minute timeout, the path for user 1 from the sample log is broken into two separate sessions since the last two references are over an hour later than the first five. The session identification step results in four user sessions consisting of A-B-F-O-G, A-D, A-B-C-J, and L-R.

*D. Feature Extraction*

The most frequent visitors are identified and stored in a vector - users {$u_1$, u2, u3,..,$u_n$}. The most frequent pages visited are identified and stored in a vector - pages {$p_1$, p2…$p_m$}. For each user $u_i$ the number of visits made to each page $p_i$ is then collected and stored in a vector called access pattern. Thus, aij in the access pattern vector indicates the number of times user i has visited the page j. Thus, we can say that this vector gives information about the preferences of each user visiting our site.

$$a_{ij} = \begin{cases} 1 & a_{ij} > x \\ 0 & \text{otherwise} \end{cases}$$

Where $a_{ij}$ - no. of visits made by $i^{th}$ user to $j^{th}$ page. Hence, each row contains the access pattern of the users visiting the site. Each row is then normalized such that,

$$\Sigma_j a_{ij} = 1$$

The normalization eliminates the effect of the magnitude of the vector during comparisons. The normalized vector is then given as input to the clustering module to group the user access patterns based on their similarities. Since, ART1 network is used for clustering purpose; the normalized user access patterns are binary encoded as follows.

$$\begin{array}{c} \\ U \\ UUs \\ Use \end{array} \begin{pmatrix} P & & & & \\ a_{11} & a_{12} & a_{13} & \ldots\ldots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \ldots\ldots & a_{2m} \\ \ldots\ldots & & & & \\ \ldots\ldots & & & & \\ \ldots\ldots & & & & \end{pmatrix}$$

U-User and P-Page, $a_{11}$ is the no. of times user $u_i$ have been accessing the page $p_i$.

Here an approach for the access pattern matrix generation.

***//Algorithm for Access Matrix Generation***

***//Input:*** *Integer Matrix A and threshold T*

***//Output:*** *Boolean Access Matrix A*

Step 1: Calculate sum of each row that is

i.e. Tot=$\sum$A1j

Step 2: Divide each row element of the matrix

by corresponding row sum that is Tot.

Step 3: if each entry in the matrix is greater

than the threshold value T then replaces

it by '1' else replaces it by '0'

From the above matrix 1 represents the page is most frequently accessed by particular user and 0 represents the not frequently accessed.

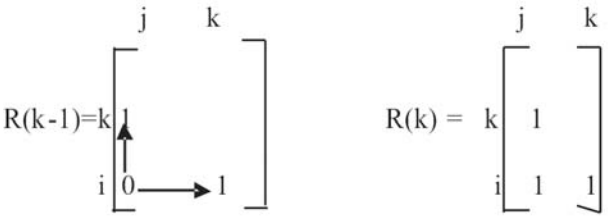
The above algorithm eliminates dynamically changing behavior of the user and gives the most frequently and recently accessed pages for clustering.

***//Algorithm for knowing hidden information from access Matrix***

The transitive closure of a directed graph is the Boolean matrix that has 1 in its $i^{th}$ row and $j^{th}$ column if and only if there is a directed edge from $i^{th}$ vertices to $j^{th}$ vertices. Warshall algorithm constructs the transitive closure of a given matrix through n-by-n Boolean matrixes.

$R^{(0)\ldots\ldots}, R^{(k-1),} R^{(k),\ldots.} R^{(n)}$

Each of these matrices provides certain information about directed path in the diagraph. specifically, the element $r_{i,j}^{(k)}$ in the $i^{th}$ row and $j^{th}$ column of matrix $R^{(k)}$ (k=0,1,….,n) is equal to 1 if and only if there exists a directed path from the $i^{th}$ vertices to $j^{th}$ vertices with each intermediate vertex, if any number not higher than k .Thus, the series starts with $R^{(0)}$ which does not allow any intermediate vertex in its path, hence $R^{(0)}$ is nothing else but the adjacency matrix of the digraph.




$r_{ij}^{(k)}=r_{ij}^{(k-1)}$ or $r_{ik}^{(k-1)}$ and $r_{kj}^{(k-1)}$ -----> [1]

Formula (1) is at heart of Warshall algorithm. This formula implies the following rule for generating elements of matrix $R^{(k)}$ from elements of matrix $R^{(k-1)}$,which is particularly convenient for applying Warshall algorithm by hand:

- if an element $r_{ij}$ is 1 in R(k-1),it remains 1 in $R^{(k)}$.
- if an element $r_{ij}$ is 0 in $R^{(k-1)}$, it has to be changed to in $R^{(k)}$ if and only if the element in its row I and column k and the element in its column j and roe k are both 1's in $R^{(k-1)}$.

**Algorithm Warshall (A[1..n,1..n])**

//Implement Warshall Algorithm for computing the Transitive Closure

//Input: The Adjacency Matrix A of a digraph with n Vertices

//Output: Transitive closure of a matrix

$R^{(0)}$ <-A

**for** k<-1 to n do

  **for** i<- 1 to n do

    **for** j <- 1 to n do

      $R^{(k)}[i,j]$ <- $R^{(k-1)}[i,j]$ **or** $R^{(k-1)}[i,k]$

        **and** $R^{(k-1)}[k,j]$

**return** $R^{(n)}$

Its time efficiency is only in O ($n^3$).In fact for sparse graph represented by their adjacency linked lists; the traversal based algorithm has better efficiency than Warshall algorithm. Space efficiency of Warshall algorithm is poor, because it require extra memory for storing the intermediate matrix.

## IV. CONCLUSIONS

This paper has presented the details of preprocessing tasks that are necessary for performing Web Usage Mining, the application of data mining and knowledge discovery techniques to WWW server access logs. This paper also presented experimental results on synthetic data for the purpose of comparing transaction identification approaches, and on real-world industrial data to illustrate some of its applications. The transactions identified with the reference length approach performed consistently well on both the real data and the created data. For the real data, only the reference length transactions discovered rules that could not be reasonably inferred from the structure of the Web sites. Since the important page in a traversal path is not always the last one, the content-only transactions identified with the maximal forward reference approach did not work well with real data that had a high degree of connectivity. The auxiliary-content transactions led to an overwhelmingly large set of rules, which limits the value of the data mining process. Future work will include further tests to verify the user browsing behavior model discussed in Sect. 4 and a more rigorous analysis of the shape of reference length

histograms in order to refine the reference length transaction identification approach.

## V. ACKNOWLEDGMENT

Special acknowledgements to Prof. Dr.V.Murali Bhaskaran,Prof. Dr.A.M.Natarajan and Dr. Illango Krishnamurthy for the insightful comments and suggestions.

## VI. REFERENCES

[1]. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of the 20th VLDB Conference, pages 487{499, Santiago, Chile, 1994.

[2]. T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible markup language (XML) 1.0 W3C recommendation. Technical report, W3C, 1998.

[3]. M. Balabanovic and Y. Shoham. Learning information retrieval agents: Experiments with automated Web browsing. In On-line Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments, 1995.

[4]. R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the World Wide Web. In International Conference on Tools with Arti_cial Intelligence, pages 558{567, Newport Beach, CA, 1997.

**Sathiyamoorthi .V,** is a Research Scholar in Data Mining at Anna University, Coimbatore. His main research interests are Database, Web Mining and Web information extraction. He has published many papers in various National and International Conferences.